Secure FAST: Security Enhancement in the NATO Time Sensitive Targeting Tool

Dr Orhan Cetinkaya, Dr Yakup Yildirim and Mr Michel Fortier
NATO C3 Agency
Oude Waalsdorperweg 61, 2597 AK The Hague
NETHERLANDS

{orhan.cetinkaya, yakup.yildirim, michel.fortier}@nc3a.nato.int

ABSTRACT

*Time Sensitive Targeting is a microcosm of military operations that employs the full range of resources in order to efficiently detect, identify, target, engage and assess emerging targets in the Area of Responsibility. Due to the high importance assigned to Time Sensitive Targets and the normally very compressed timelines involved in successfully engaging them, a computer-based tool to support access to critical data and the exchange of information between personnel participating in the Time Sensitive Targeting process is essential. The **F**lexible, **A**dvanced C2 Services for NATO (Joint) Time **S**ensitive **T**argeting (FAST) tool has been implemented to provide these capabilities as a coordination and collaboration tool, designed to aid in the tracking and prosecuting of Time Sensitive Targets. The FAST tool provides user level authentication and authorisation in terms of security. It utilizes operating system level security but does not provide application level security for the stored and communicated data between participants. It would be better to enhance the security of the FAST tool in terms of data and communication. This paper illustrates how the security enhancement for the FAST tool can be achieved from a theoretical perspective.*

1.0 INTRODUCTION

Time-Sensitive Targets (TSTs) are defined as those targets requiring an immediate response because they pose (or will soon pose) a danger to friendly operations or are highly lucrative, fleeting targets of opportunity [1]. Some examples of potential TSTs could include mobile C2 vehicles and facilities, deployed theatre ballistic missiles (TBMs), mobile rocket launchers (MRLs), mobile high threat Surface-to-Air Missile systems (SAMs), military or civilian individuals who pose a threat and demand an immediate action to neutralize, previously unidentified C2 nodes, terrorist leadership and mobile radio/TV broadcast stations.

Time Sensitive Targeting (TST) is a process [1] based on previously determined information that can be stored in readily accessible databases for immediate retrieval. The Flexible, Advanced C2 Services for NATO (Joint) Time Sensitive Targeting (FAST) tool [2] supports the TST process with state of the art in information technology data exchange mechanisms involving the exchange of information between non-collocated personnel. The FAST provides user level authentication and authorisation in terms of security. It uses operating system level security but does not provide application level security for the stored and communicated data between participants. It would be better to improve the security of the FAST tool in terms of data and communication. This paper explains the current architecture of the FAST tool and illustrates how security enhancements to the FAST tool can be achieved from a theoretical perspective.

2.0 FAST ARCHITECTURE

The FAST tool has been implemented to support access to critical data and the exchange of information between personnel participating in the TST process as a coordination and collaboration tool. The central

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2010	**N/A**	**-**

4. TITLE AND SUBTITLE
Secure FAST: Security Enhancement in the NATO Time Sensitive Targeting Tool

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NATO C3 Agency Oude Waalsdorperweg 61, 2597 AK The Hague NETHERLANDS

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)
11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADA564697. Information Assurance and Cyber Defence (Assurance de l'information et cyberdefense). RTO-MP-IST-091

14. ABSTRACT
Time Sensitive Targeting is a microcosm of military operations that employs the full range of resources in order to efficiently detect, identify, target, engage and assess emerging targets in the Area of Responsibility. Due to the high importance assigned to Time Sensitive Targets and the normally very compressed timelines involved in successfully engaging them, a computer-based tool to support access to critical data and the exchange of information between personnel participating in the Time Sensitive Targeting process is essential. The Flexible, Advanced C2 Services for NATO (Joint) Time Sensitive Targeting (FAST) tool has been implemented to provide these capabilities as a coordination and collaboration tool, designed to aid in the tracking and prosecuting of Time Sensitive Targets. The FAST tool provides user level authentication and authorisation in terms of security. It utilizes operating system level security but does not provide application level security for the stored and communicated data between participants. It would be better to enhance the security of the FAST tool in terms of data and communication. This paper illustrates how the security enhancement for the FAST tool can be achieved from a theoretical perspective.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**6**	

part of the FAST, the Joint Time Sensitive Targeting (JTST) dashboard, has been designed to present the large amount of data pertaining to the current state of target prosecution in a condensed, yet easy-to-understand, way. Communication within the cell can be a problem due to the number of participants in a cell, the remoteness of some participants and the time sensitivity of the target. To streamline the communication process, the FAST tool has a built-in chat client. This client allows participants to communicate with each other and possibly with other cells. This communication is visible to all parties involved, so duplicate discussions amongst different cell members can be avoided.

The FAST tool is used throughout all levels of Command in NATO. The coordination within the Joint TST Cell is supported in a multi-site networked and multi-service Command and Control (C2) structure. These positions interact through the Cell's local area network, and a wide area network provides connectivity between the JTST Cell and the components.

The FAST tool setup consists of a FAST Server and local FAST workstations. It is required that all participants be connected to a FAST server which acts as a message gateway allowing the required data to be shared amongst all participants at all times. Once the data has been created, it must be distributed to all participants.

The FAST data is stored on the FAST server and each of the local FAST workstations in XML files (Figure 1). The workstations access their own data (every participant normally has the same dataset because any data change made by one of the participants is automatically distributed to others). Any action (TST prosecution) is stored and can be reached by the members joining the TST cell at some point and for subsequent legal auditing if required.

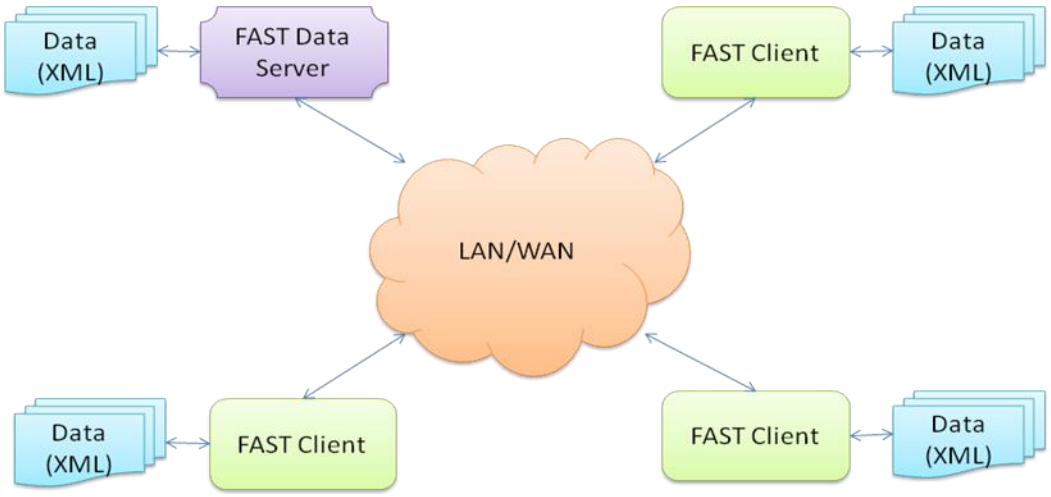

Figure 1: FAST Architecture

Because of FAST requirements, a solution has been chosen without a central database. All FAST users do connect to a central FAST Data Server to share data. The server itself does not have any data state; it merely acts as a data relay station between different FAST users. The server is placed somewhere on a Local Area Network (LAN) or Wide Area Network (WAN).

As there is no central storage on the network, all FAST clients maintain their own system state (in the form of XML files). The FAST application makes sure that all users will have the same state at all times when they are connected to the server. One of the advantages of not having central data storage is that in case of any malfunctioning of the TST Data Server, it is only a matter of starting another server at another machine, or even at another site. Since the server has no data state, the operations can in this case continue

immediately. In future FAST versions, it might be possible to have an automatic fall-over to another predefined backup server. In that case, the user will not notice that the original server went down. In order to better understand FAST data exchange architecture and mechanisms, some key operational details will be stated briefly.

- A complete history of all the transactions in the tasking process (what was tasked by who and when) is kept in the target XML file.
- To make it impossible to modify the file, a checksum is introduced for each file, so that if it has been modified, the modification can be detected and the file can be rejected.
- Critical information in the XML files, such as passwords, is encrypted.
- Access to FAST requires the use of passwords.
- Some parts of the system state are specific within a component, for example Target data. FAST only allows you to see/modify component specific data if you are logged-on to the right component.
- To prevent users from taking any role in the component in the Component Setup Tool each role by default has to be provided with a role-based password, but the Administrator has the option to allow login with only the component password as well.
- If, for whatever reason, a user is not happy with the system state they can request the FAST network to update the system state for them. The user has the option to request a complete new system state (all XML files), or only a part of the system state.
- As with all user actions within FAST the delete action is instantaneously and automatically transferred to all other connected TST users and observers. But before the TST Target is actually deleted from the FAST system state a full time-stamped backup of the target data XML is made, so that the deletion can always be reversed.

The FAST tool client software is supported on Microsoft Windows 2000, Windows XP and Windows Vista operating systems. The Windows PC is the primary target platform for the FAST tool client software. However, FAST can also run on Solaris 10 x86 and Solaris 10 SPARC operating systems.

The FAST tool server components are supported on the Windows and Solaris 10 operating systems. Supported FAST tool server hardware includes both Sun SPARC and x86 PC. Windows XP and Vista are not servers, but they can run the FAST application as well.

3.0 SECURITY ENHANCED FAST ARCHITECTURE

The overall security enhanced FAST architecture assumes that a NATO-wide Public Key Infrastructure (NATO PKI) is available and that the FAST server and every client has public-private key pairs. The following notation will be used:

S: Server
C: Client
$\mathbf{\{M\}_{SC}}$: Encrypt message M with S and C's symmetric key
$\mathbf{\{M\}_{S}}$: Encrypt message M with S's public key
$\mathbf{[M]_{S}}$: Sign the message M with S's private key
$\mathbf{\{M\}_{SK}}$: Encrypt message M with session key SK
$\mathbf{\{M\}_{DK}}$: Encrypt message M with data key DK

In order to keep data in a more secure way, the data is required to be encrypted instead of being stored as plain text. In the new architecture, symmetric key cryptography is used to maintain data security. The key used for the data encryption is referred to as the "Data Key". All the data is kept in an encrypted form using the Data Key.

The server generates the Data Key and shares it with the clients, so all the clients use the same key for the consistency of the databases. In theory, there is no obvious reason to prevent the use of different data keys for every participant. However, it is better to use one key for every database for maintenance and backup purposes. The server signs the Data Key with its private key ([Data Key]S) and encrypts the signed result with the relevant client's public key and sends the message ({[Data Key]S}C) to the client. When the client receives the message, it decrypts the message with its private key and applies the server's public key and gets the Data Key. The same process is applied for each client. Eventually all the clients have the same Data Key. If the server changes the Data Key, it should apply the same procedures to distribute the key to the clients. (Figure 2)

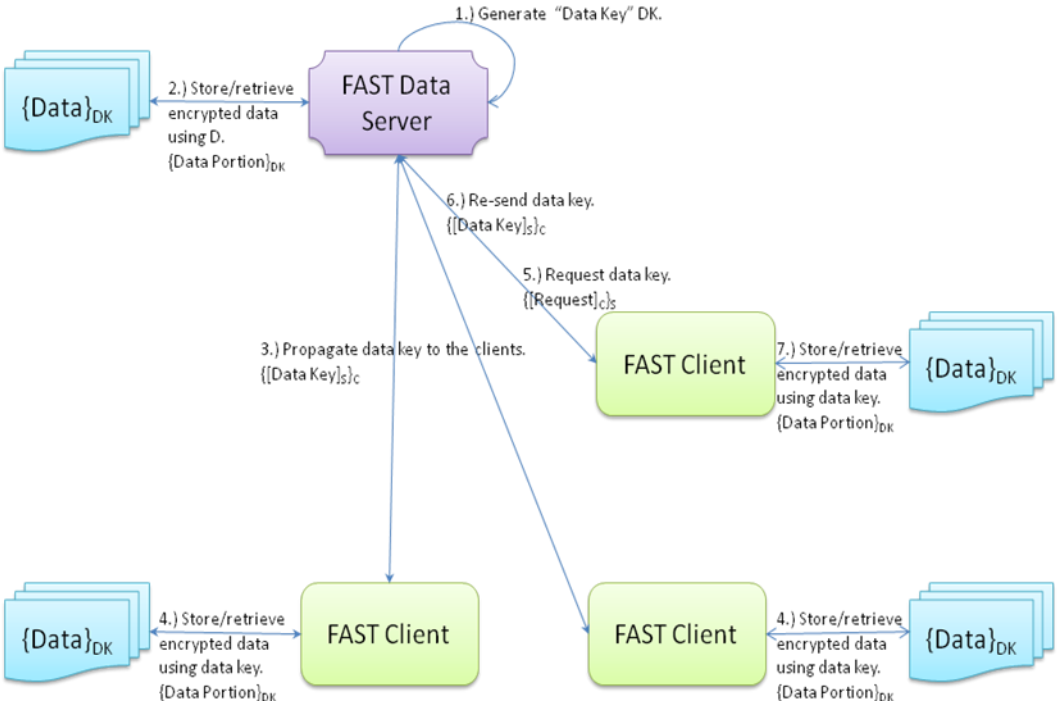

Figure 2: FAST Data Key Establishment

Communication security can also be enhanced using the encrypted message exchange mechanism instead of plain text. In order to do this, a session key is needed when the clients wants to talk with the server. The FAST has a distributed architecture. In other words, there are many clients who can communicate with and send data to the server. In this case, there are two options: The server can use different session keys for every client or one session key can be used for all clients. The former has higher maintenance complexity since key management is needed for each client, thus the latter is utilized.

The server generates a "Session Key" to enable secure communication within the clients, and stores the key until the session ends. All communications are carried out using this Session Key. If any client joins the network, it should first request the current Session Key from the server before starting communications. When the server changes the Session Key, it should propagate the key to all of the clients. The Session Key exchange is carried out in a similar way to the Data Key exchange mechanism.

The server signs the Session Key with its private key ([Session Key]S) and encrypts the signed result with the relevant client's public key and sends the message ({[Session Key]S}C) to the client. The clients can easily extract the Session Key by using their private key and server's public key (Figure 3).

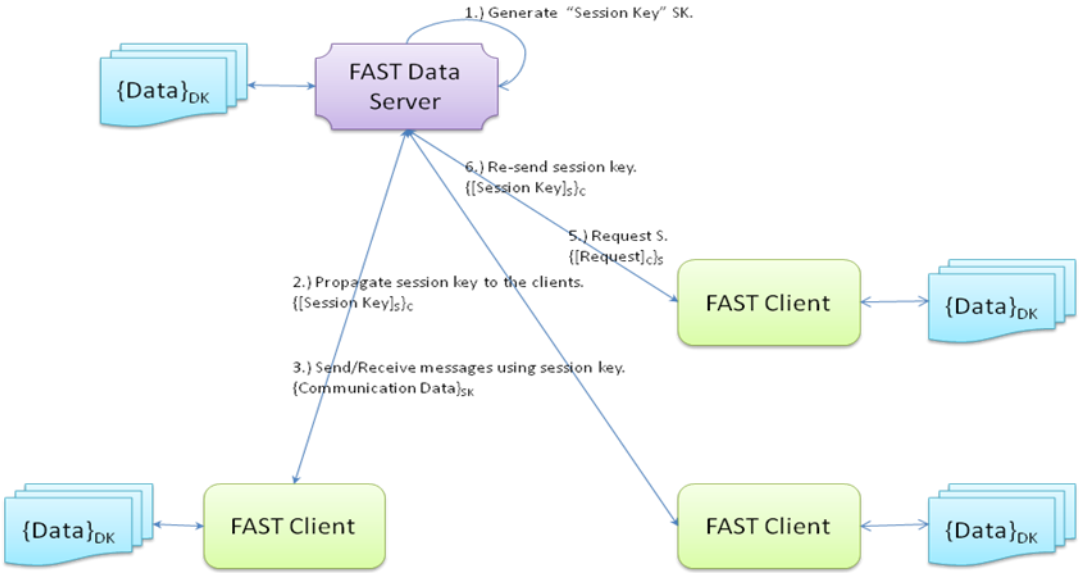

Figure 3: FAST Session Key Establishment

The server stores both the data and session keys. It may continue to use the old keys after restarting. In this way it does not have to generate the keys for every restart. Every client carries out an initiation task for first time use. When a client joins the network, it should check its keys against to the server. If the keys have been changed, it should request the latest keys. If the server decides to change the keys, it should propagate the new keys to the clients. Thus the server and all the clients always have the same data and session keys.

4.0 CONCLUSION

The FAST tool has been implemented as a coordination and collaboration tool to support access to critical data and the exchange of information between personnel participating in the TST process as coordination and collaboration tool, designed to aid in the tracking and prosecuting of TSTs. It should be noted that the important role that operators played in the requirements capture process and that the intent of the tool is to serve as the first step in a spiral development process which will involve industry at a later stage. Furthermore, the FAST tool can be adapted to support processes such as dynamic targeting and search and rescue operations where coordination and collaboration is important. The current FAST version does not provide data and communication protection against hostile attacks. The security enhancements described in this paper resolve this and protect the FAST data and communications traffic.

5.0 REFERENCES

[1.] *"AJP-3.9, Allied Joint Doctrine for Joint Targeting"*, NATO Standardization Agency (NSA), BE, May 2008 (NU)

[2.] Y. Yildirim, O. Cetinkaya and M. Fortier, "*Coordination and Collaboration in Time Sensitive Targeting*", In Proceedings of Military Communications and Information Systems Conference (MCC2009), Prague, Czech Republic, 29-30 September 2009 (NU)